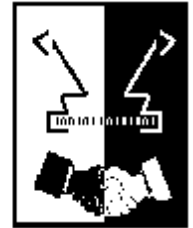


SECURE

Secure Environments for Collaboration
among Ubiquitous Roaming Entities



Trust as Evidence of Behaviour

Karl Krukow, BRICS,
University of Aarhus, Denmark

International Workshop on
Global Computing, March 10, 2004,
Rovereto





Overview

- SECURE: abstract overview
- An evidence-based approach to trust
 - Illustrative example
- The model for trust in SECURE
- Summary





SECURE

- The essence of SECURE:
 - Explore to which extent the (intuition behind) human notion of trust can be exploited to enable security related decision-making for computational entities in a global computing environment.
- Intuition behind the *human notion of trust*:
 - Trust exists *between principals*.
 - Reflects the behaviour of principals.





SECURE

- *Trust-based, security-related* decisions:
 - Passive: e.g. should I allow principal P to access my resource r?
 - Active: e.g. which of P, Q, R will provide the best service for me?
- Properties of a *GC environment* implies that:
 - Collaboration among entities is necessary to obtain goals.
 - Entities are mobile and networked, but decisions must be made autonomously.
 - It is not possible to have complete information about entities with whom interaction is considered.





More Concretely

- A decision involving another entity P has have a number of potential outcomes

O_1, O_2, \dots, O_n

- Each outcome has an associated cost/benefit, $\text{cost}(O_i)$

$$\text{exp} = \sum_i \text{cost}(O_i) * \text{likelihood}(O_i)$$

- Trust information must support this!





Modeling Trust

- Abstractly one might model trust as simply an element of a set T of 'trust values'.
- `trust-state`: $P \rightarrow P \rightarrow T$
 - `trust-state(A)(B)`: represents A's trust in B
- Stephen Weeks: Understanding Trust Management Systems, 2001
 - T is a lattice where ordering represents more trust.
 - `trust-state` is the least fixed point of principal trust policies.
 - Essentially a distributed form of access-control.





Evidence Based Approach to Trust

- An arbitrary complete lattice is too abstract for our purposes.
 - There is no canonical way to derive estimates for the likelihood of outcomes of interaction.
- So we require additional structure:
 - The set T of trust values must explicitly represent outcomes, $T = Outcomes \rightarrow EvidenceValues$
 - *Outcomes* and *EvidenceValues* also has more structure...





Example: E-Purse

- Consider a GC-like scenario where each entity can store some amount of electronic cash in an electronic "purse".
- Entities may transfer e-money from one e-purse to another.
- Users may transfer "real" money from their bank accounts to e-cash stored in e-purses.
- Now consider a scenario where a user is considering requesting an amount m of e-cash from a bank.





Example: E-Purse

- Seen from the point of view of the user there are various possible events that may occur
 - The request may be denied for several reasons
 - Because of insufficient funds in user account
 - Because the bank server is down for maintenance
 - Because there is no reply within a certain time limit
 - The request may be granted - the bank transfers m units
 - The bank may withdraw an amount different from m from users account
 - The bank may withdraws the correct amount
 - The transferred e-money may be forged
 - The transferred e-money may be authentic
 - ...





Observations on Observations

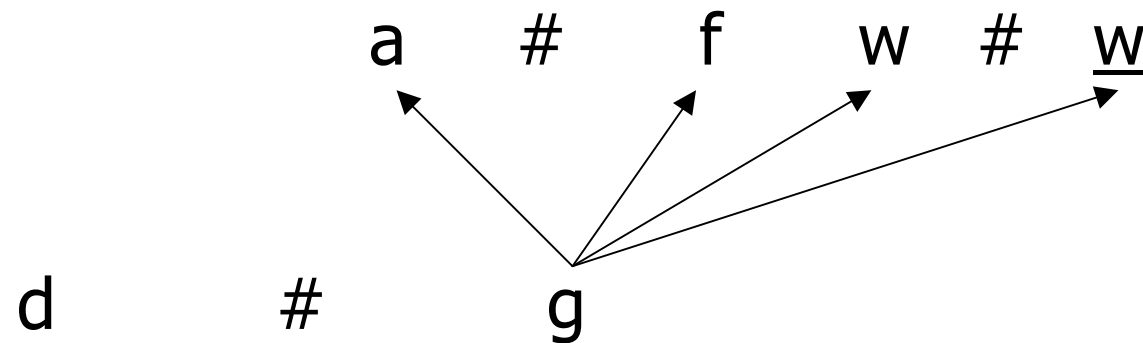
- Observations may be in conflict:
 - For example the observation of "granted" excludes the observation of "denied" since both can't occur within the same transaction.
- Observations may be dependent:
 - For example an observation of "forged" money only makes sense in a scenario where the transfer was "granted".
- Observation may be in-dependent:
 - The observation of withdrawal on the bank account and whether or not the money is forged can be made independently in any order.





Modeling (part of) the E-Purse scenario

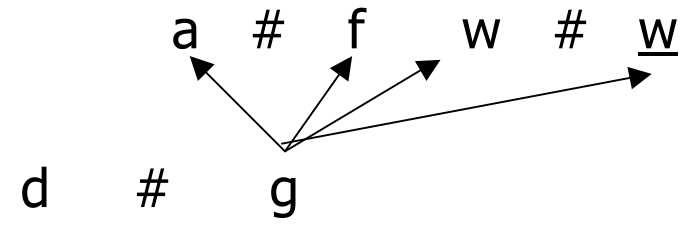
- One can model the possible observations as an event structure
 - Formally a set E of events and two relations $\#$ (conflict) and \rightarrow (causality or necessity) + some properties



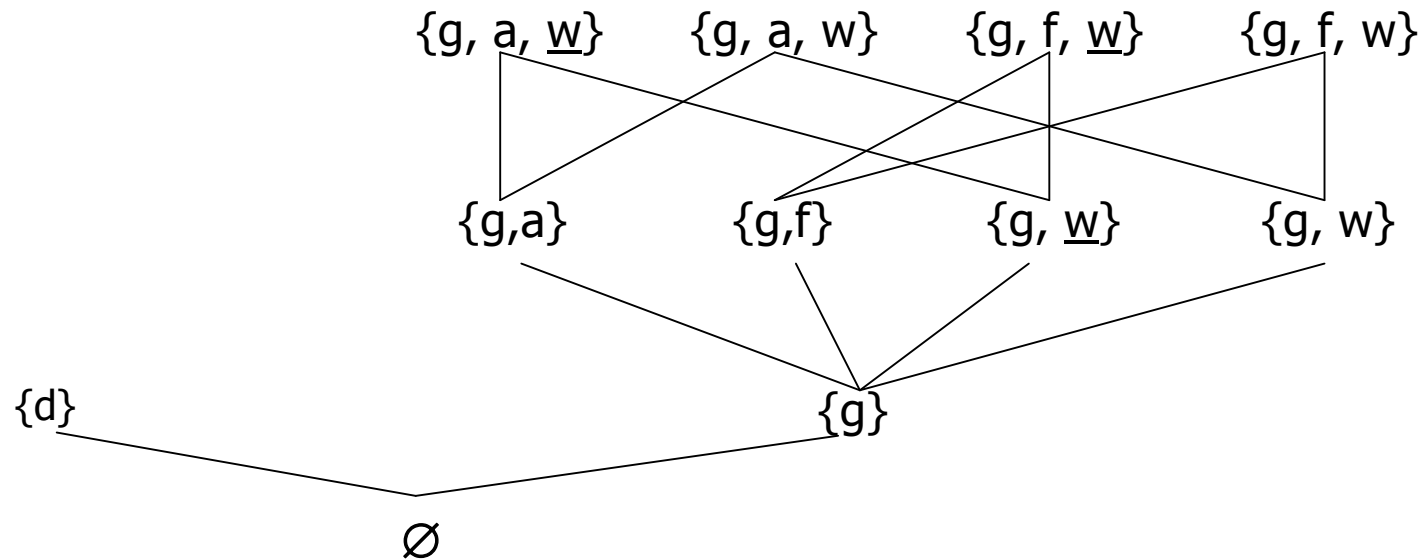


Modeling E-Purse: Outcomes

- For event structure:



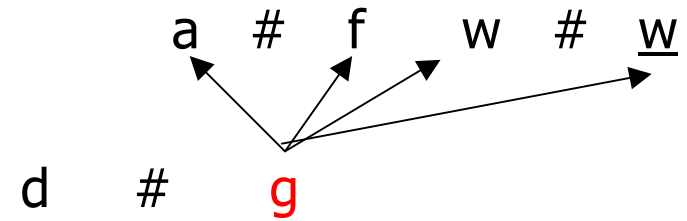
- Configurations model the information a principal has about an interaction



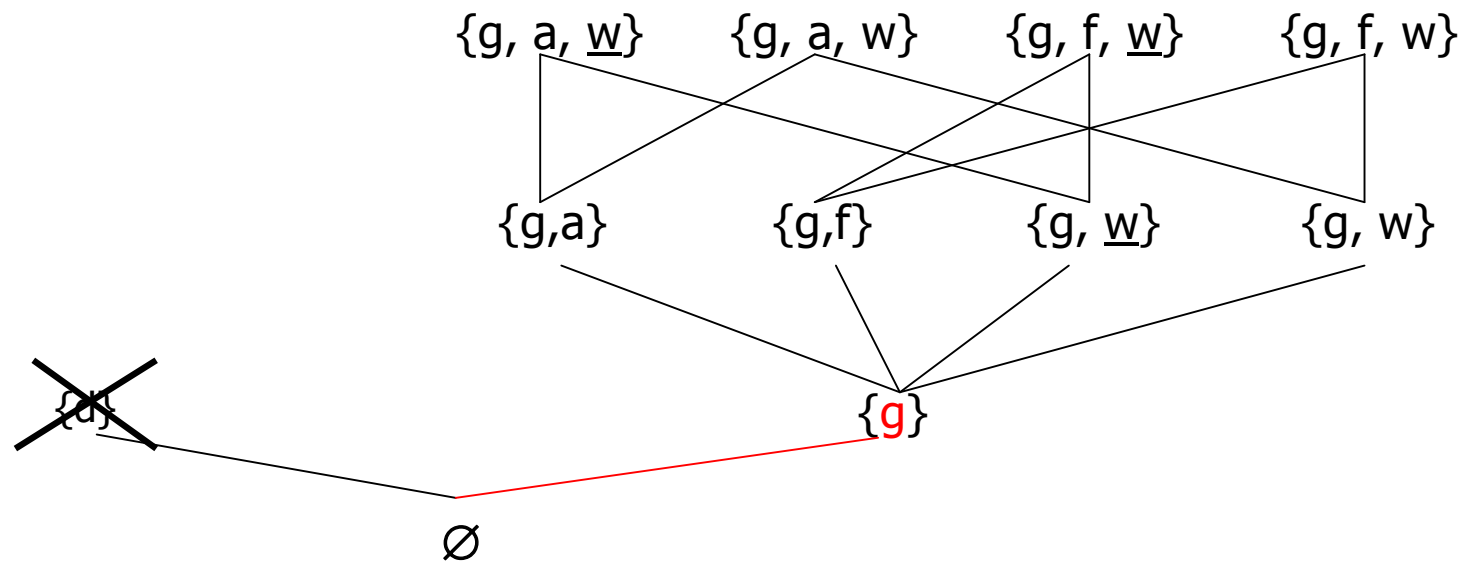


Modeling E-Purse: Monitoring Interaction

- For event structure:



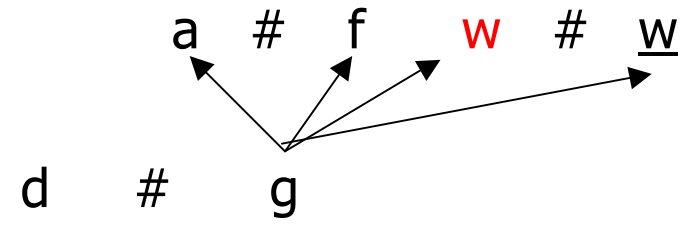
- Observe event g



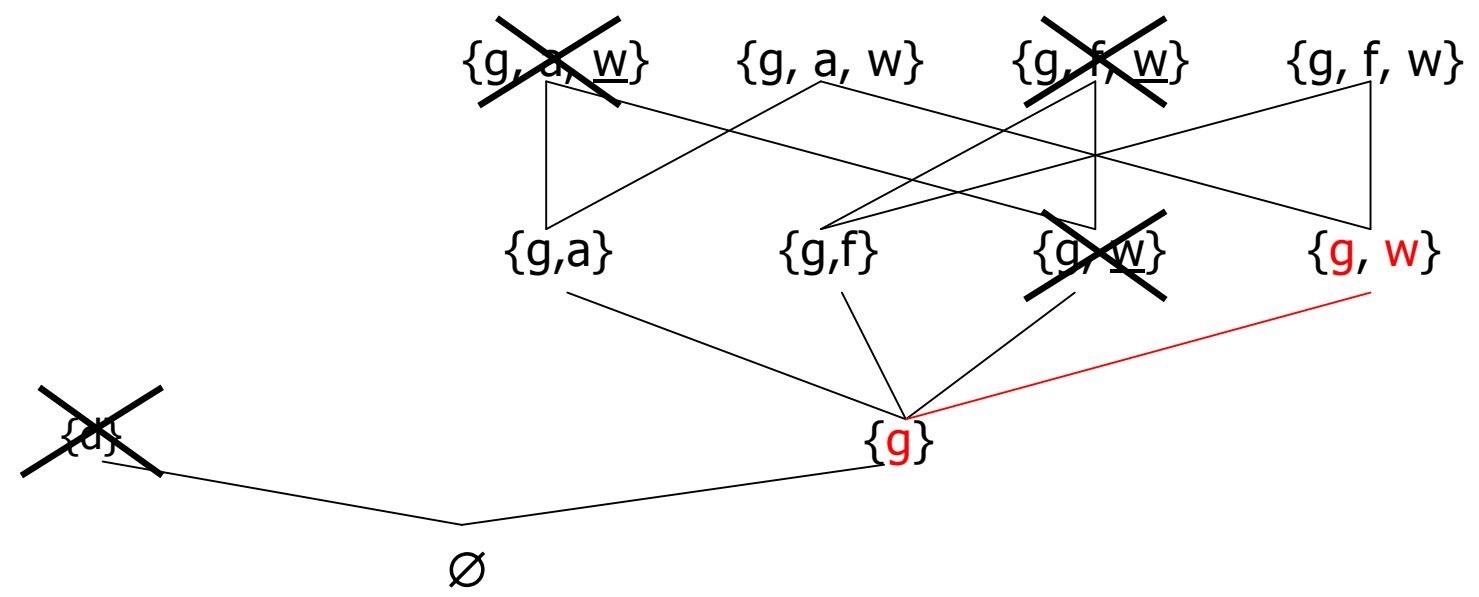


Modeling E-Purse: Monitoring Interaction

- For event structure:



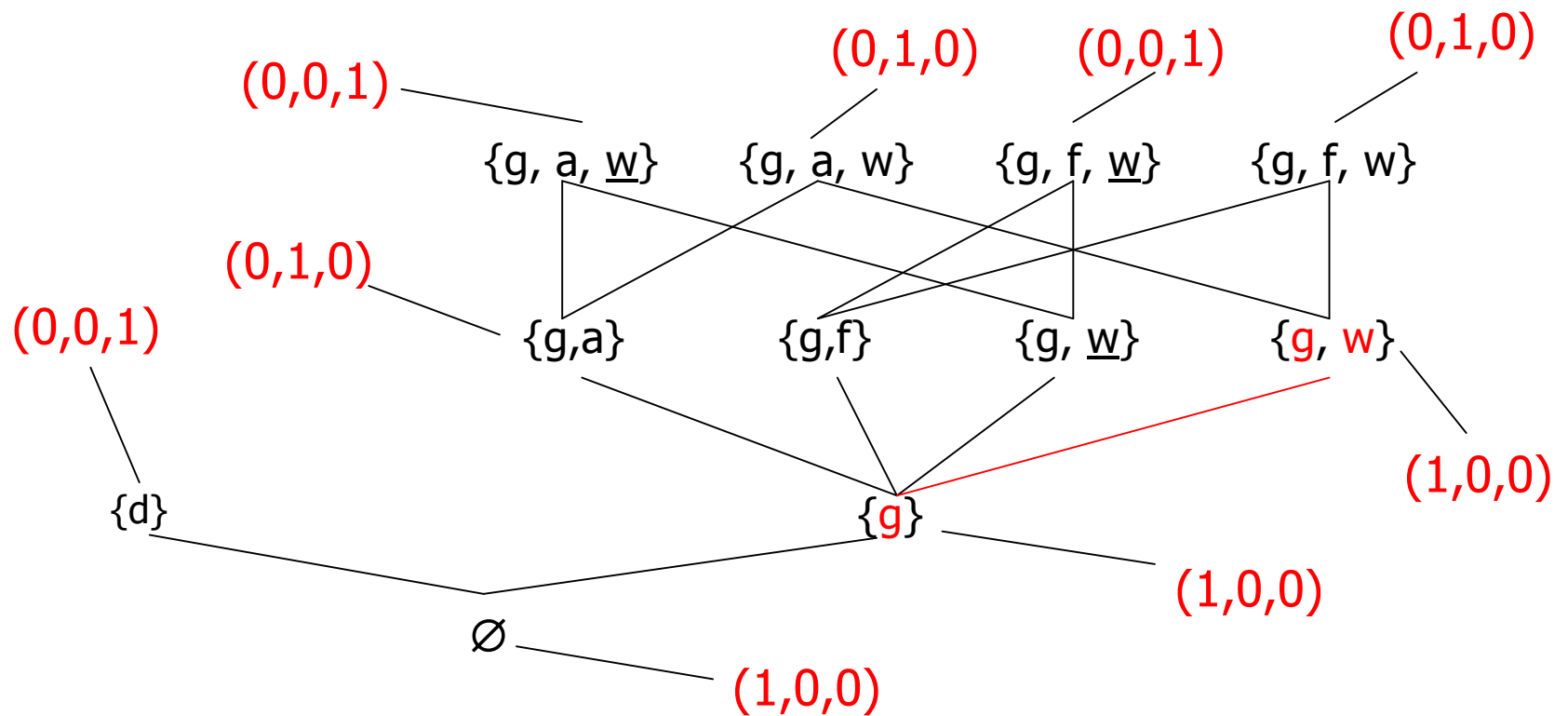
- Observe event **w**





Modeling E-Purse: Deriving Values

- Deriving values





Generally

- The general approach:
 - To model each *decision* by an event structure
 $ES = (E, \leq, \#)$
 - Trust values are functions $t \in T = \text{FinConf}(ES) \rightarrow \mathbb{N}^3$
- Each principal maintains an interaction history for other principals:
 - A sequence, $H \in \text{FinConf}(ES)^*$, where each configuration h_i in H models the information about a particular interaction, and $|H|$ is the total number of interactions.



Deriving Trust Values

- We can transform such an H into a piece of trust information
 - $\text{eval}(H): \text{FinConf}(\text{ES}) \rightarrow \mathbb{N}^3$ (local trust information)
- $\text{eval}(H)(o) = (s,i,c)$ means out of $s+i+c$ interactions
 - s : interactions supports o
 - i : interactions were inconclusive about o
 - c : interactions contradicted o





An information ordering on \mathbb{N}^3

- We can define an information ordering on \mathbb{N}^3 :
 - $(s,i,c) \sqsubseteq (s',i',c')$ iff
 - $s \leq s'$, $c \leq c'$ and $s + i + c \leq s' + i' + c'$
- Adjoining a top element makes $(\mathbb{N}^3, \sqsubseteq)$ a complete lattice.
- This ordering lifts (point-wise) to the function space $\text{FinConf}(\text{ES}) \rightarrow \mathbb{N}^3$
- On derived values ($\text{eval}(H)$), the order \sqsubseteq corresponds to either refining or adding new interactions some number of times.



Trust Policies

- Each principal P defines a trust policy π_A
 - Defines how A computes its trust in any other principal.
- A policy language could have constructs like
 - Refer to the information gathered locally
 - Refer to information that principal P has personally observed
 - Refer to the information P would obtain if it were to compute its trust
 - Other monotone operations...





Trust Policies – Semantics

- How to resolve reference cycles?
 - A's trust in P is B's trust in P
 - B's trust in P is A's trust in P
 - Intuitively this value should be (0,0,0)!
- Semantics of a policy

$$\llbracket \pi_A \rrbracket : \underbrace{(P \rightarrow P \rightarrow O \rightarrow \mathbb{N}^3)}_{\text{Trust-State}} \rightarrow \underbrace{P \rightarrow O \rightarrow \mathbb{N}^3}_{\text{A's trust}}$$

- i.e. A's policy could be $\lambda m. \lambda X. m(B)(X)$





Trust Policies – Semantics

- For any collection of mutually referring policies, $\langle \pi_A : A \in P \rangle$
- Unique Trust-State

$$\langle \pi_A : A \in P \rangle : \underbrace{(P \rightarrow P \rightarrow O \rightarrow \mathbb{N}^3)}_{\text{Trust-State}} \rightarrow \underbrace{P \rightarrow P \rightarrow O \rightarrow \mathbb{N}^3}_{\text{Trust-State}}$$

$$\llbracket \langle \pi_A : A \in P \rangle \rrbracket = \text{lfp } \langle \pi_A : A \in P \rangle$$



Summary

- SECURE projects uses intuition behind the human notion of trust in its trust model.
 - In SECURE, cost/benefit, risk, outcomes, observation are modeled explicitly.
- Event Structures seem to model well the notion of observations and outcomes.
 - Allows for defining and deriving trust values that enable reasoning about likelihood of outcomes.
- Principals specify how they define their trust with trust-policies:
 - At any time a collection of such mutually referring policies uniquely define a trust state.





Trust as Evidence of Behaviour

Thank You !

